Московский государственный технический

университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по Рубежному контролю №1

Выполнил: Проверил:

студент группы ИУ5-34Б преподаватель кафедры ИУ5

Григорян А.А. Гапанюк Ю.Е.

Подпись и дата: Подпись и дата:

Москва, 2021г.

**Постановка задачи:**

Вариант Г5

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.

3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

В предметной области в данном варианте класс «Отдел» соответствует классу «Оркестр», а класс «Сотрудник» - классу «Музыкант»

**Текст программы:**

import operator

class Mus:

def \_\_init\_\_(self, id, fio, sal, orch\_id):

self.id = id

self.fio = fio

self.sal = sal

self.orc\_id = orch\_id

class Orch:

def \_\_init\_\_(self, id, name):

self.id = id

self.name = name

class Mus\_Orch:

def \_\_init\_\_(self, mus\_id, orch\_id):

self.mus\_id = mus\_id

self.orch\_id = orch\_id

orch = [

Orch(1, "Московский симфонический оркестр"),

Orch(2, "Симфонический оркестр Большого театра"),

Orch(3, "Оркестр имени П.И.Чайковского"),

Orch(4, "Государственный оркестр России"),

Orch(5, "Академический оркестр Московской филармонии"),

]

mus = [

Mus(1, "Иванов", 67000, 5),

Mus(2, "Дмитриев", 55000, 1),

Mus(3, "Васильев", 89000, 3),

Mus(4, "Герасимов",64000, 4),

Mus(5, "Зайцев", 95000, 4),

]

mus\_orch = [

Mus\_Orch(1, 5),

Mus\_Orch(2, 1),

Mus\_Orch(3, 3),

Mus\_Orch(4, 4),

Mus\_Orch(5, 4)

]

def main():

# соединение данных один-ко-многим

one\_to\_many = [(o.name, m.fio, m.sal)

for o in orch

for m in mus

if m.orc\_id == o.id

]

# соединение данных многие-ко-многим

many\_to\_many\_temp = [(o.name, m\_o.orch\_id, m\_o.mus\_id)

for o in orch

for m\_o in mus\_orch

if o.id == m\_o.orch\_id]

many\_to\_many = [(m.fio, m.sal, orch\_name)

for orch\_name, orch\_id, mus\_id in many\_to\_many\_temp

for m in mus if m.id == mus\_id]

#------------------------Задание Г1-------------------------------------------

res1 = []

for o in orch:

if o.name[0] == 'А':

# cписок музыкантов оркестра

orch\_mus = list(filter(lambda i: i[0] == o.name, one\_to\_many))

if len(orch\_mus) > 0:

# имена сотрудников отдела

names\_mus = [name for \_,name,\_ in orch\_mus]

res1.append((o.name, names\_mus))

print("Задание Г1: ")

print(res1)

#-----------------------------------------------------------------------------

#------------------------Задание Г2---------------------------------

# список для оркестров, названия которых начинаются с буквы 'А'

res2\_unsorted = []

# зарплаты

sal = []

for o in orch:

# cписок музыкантов оркестра

orch\_mus = list(filter(lambda i: i[0] == o.name, one\_to\_many))

if len(orch\_mus) > 0:

# зарплаты сотрудников отдела

sal = [sal for \_,\_,sal in orch\_mus]

# максимальная зарплата

max\_sal = max(sal)

res2\_unsorted.append((o.name, max\_sal))

res2\_sorted = sorted(res2\_unsorted, key=operator.itemgetter(1), reverse=True)

print("Задание Г2: ")

print(res2\_sorted)

#---------------------------------------------------------------------

#--------------------------Задание Г3-------------------------------

res3\_unsorted = []

for o in orch:

# cписок музыкантов оркестра

orch\_mus = list(filter(lambda i: i[2] == o.name, many\_to\_many))

if len(orch\_mus) > 0:

fio\_mus\_unsorted = [fio for fio,\_,\_ in orch\_mus]

# отсортрованный список фамилий музыкантов

fio\_mus\_sorted = sorted(fio\_mus\_unsorted, key=operator.itemgetter(0))

res3\_unsorted.append((o.name, fio\_mus\_sorted))

res3\_sorted = sorted(res3\_unsorted, key=operator.itemgetter(0), reverse=True)

print("Задание Г3: ")

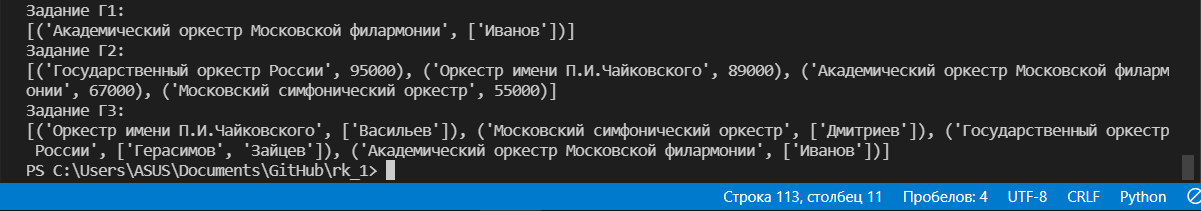
print(res3\_sorted)

#-------------------------------------------------------------------

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результат выполнения:**

****